

Code Baseline for the Land Information System
Submitted under Task Agreement GSFC-CT-2
Cooperative Agreement Notice (CAN)
CAN-00OES-01
Increasing Interoperability and Performance of
Grand Challenge
Applications in the Earth, Space, Life, and
Microgravity Sciences

Version 1.0

Draft 2

August 12, 2002

Contents

1	Description of the Milestone	4
2	Description of Algorithms	4
2.1	Land Surface Modeling and Data Assimilation	4
2.2	Land Data Assimilation System (LDAS)	4
2.3	Community Land Model (CLM)	7
2.4	The Community NOAH Land Surface Model	7
3	Description of the Test Case	8
4	Description of the Computer Code Used	8
4.1	Documentation of the Computer Code	9
4.2	Code Repository	9
5	Results	9
5.1	Total Execution Times	9
5.2	CPU Times	9
5.3	Disk Usage	15
5.4	Memory Usage	17
5.5	Computational Requirements at 1km	17
6	Conclusions	18
7	Future Directions	19

List of Figures

1	Flowchart for LDAS	6
2	Total execution times for the LDAS runs on HOPPER and LOMAX.	10
3	CPU times of computationally intensive functions on HOPPER for CLM runs	11
4	CPU times of computationally intensive functions on LOMAX for CLM runs	11
5	Percentage of total CPU times for computationally intensive functions on HOPPER for CLM runs	12
6	Percentage of total CPU times for computationally intensive functions on LOMAX for CLM runs	12

7	CPU times of computationally intensive functions on HOPPER for NOAH runs	13
8	CPU times of computationally intensive functions on LOMAX for NOAH runs	13
9	Percentage of total CPU times for computationally intensive functions on HOPPER for NOAH runs	14
10	Percentage of total CPU times for computationally intensive functions on LOMAX for NOAH runs	14
11	Effect of timestep duration on the computational times for CLM runs	15
12	Effect of domain increase from $2^\circ \times 2.5^\circ$ to $1/4^\circ$ on the computational times for CLM runs	16
13	Effect of domain increase from $2^\circ \times 2.5^\circ$ to $1/4^\circ$ on the computational times for NOAH runs	16

List of Tables

1	Measure of computational intensity for LDAS $1/4^\circ$ runs (ms/gridcell/day)	17
2	Disk Usage for various LDAS runs (in MB)	17
3	Memory Usage for various LDAS runs	17

1 Description of the Milestone

The milestone for the Land Data Assimilation System (LDAS) [5] code baseline deals with the implementation and execution of the Community Land Model (CLM) [2] and the National Oceanic and Atmospheric Administration's NOAA (National Center for Environmental Prediction, Oregon State University, United States Air Force, and Office of Hydrology) [6] land surface model (LSM) within the LDAS at 1/4° resolution on the ESS Testbed for the near-term retrospective period. The milestone also requires publishing an initial version of documented source code made publicly available via the Web. The expected completion date is July 2002.

2 Description of Algorithms

This section provides an overall description of the land surface modeling and data assimilation, followed by a description of the algorithms for each individual components of LIS involved in this baselining study.

2.1 Land Surface Modeling and Data Assimilation

In general, land surface modeling seeks to predict the terrestrial water, energy and biogeochemical processes by solving the governing equations of the soil-vegetation-snowpack medium. Land surface data assimilation seeks to synthesize data and land surface models to improve our ability to predict and understand these processes. The ability to predict terrestrial water, energy and biogeochemical processes is critical for applications in weather and climate prediction, agricultural forecasting, water resources management, hazard mitigation and mobility assessment.

In order to predict water, energy and biogeochemical processes using (typically 1-D vertical) partial differential equations, land surface models require three types of inputs: 1) initial conditions, which describe the initial state of land surface; 2) boundary conditions, which describe both the upper (atmospheric) fluxes or states also known as "forcings" and the lower (soil) fluxes or states; and 3) parameters, which are a function of soil, vegetation, topography, etc., and are used to solve the governing equations.

2.2 Land Data Assimilation System (LDAS)

LDAS is a model control and input/output system (consisting of a number of sub-routines, modules written in Fortran 90 source code) that drives multiple offline one dimensional land surface models (LSMs) using a vegetation defined "tile" or "patch"

approach to simulate sub-grid scale variability. The one-dimensional LSMs such as CLM and NOAH, which are subroutines of LDAS, apply the governing equations of the physical processes of the soil-vegetation-snowpack medium. These land surface models aim to characterize the transfer of mass, energy, and momentum between a vegetated surface and the atmosphere.

LDAS makes use of various satellite and ground based observation systems within a land data assimilation framework to produce optimal output fields of land surface states and fluxes. The LSM predictions are greatly improved through the use of a data assimilation environment such as the one provided by LDAS. In addition to being forced with real time output from numerical prediction models and satellite and radar precipitation measurements, LDAS derives model parameters from existing topography, vegetation and soil coverages. The model results are aggregated to various temporal and spatial scales, e.g., 3 hourly, $1/4^\circ$.

Figure 1 shows the algorithmic steps involved in LDAS. The execution of LDAS starts with reading in the user specifications. The user selects the model domain and spatial resolution, the duration and timestep of the run, the land surface model, the type of forcing from a list of model and observation based data sources, the number of "tiles" per grid square (described below), the soil parameterization scheme, reading and writing of restart files, output specifications, and the functioning of several other enhancements including elevation correction and data assimilation.

The system then reads the vegetation information and assigns subgrid tiles on which to run the one-dimensional simulations. LDAS runs its 1-D land models on vegetation-based "tiles" to simulate variability below the scale of the model grid squares. A tile is not tied to a specific location within the grid square. Each tile represents the area covered by a given vegetation type.

Memory is dynamically allocated to the global variables, many of which exist within Fortran 90 modules. The model parameters are read and computed next. The time loop begins and forcing data is read, time/space interpolation is computed and modified as necessary. Forcing data is used to specify boundary conditions to the land surface model. The LSMs in LDAS are driven by atmospheric forcing data such as precipitation, radiation, wind speed, temperature, humidity, etc., from various sources. LDAS applies spatial interpolation to convert forcing data to the appropriate resolution required by the model. Since the forcing data is read in at certain regular intervals, LDAS also temporally interpolates time average or instantaneous data to that needed by the model at the current timestep. The selected model is run for a vector of "tiles", intermediate information is stored in modular arrays, and output and restart files are written at the specified output interval.

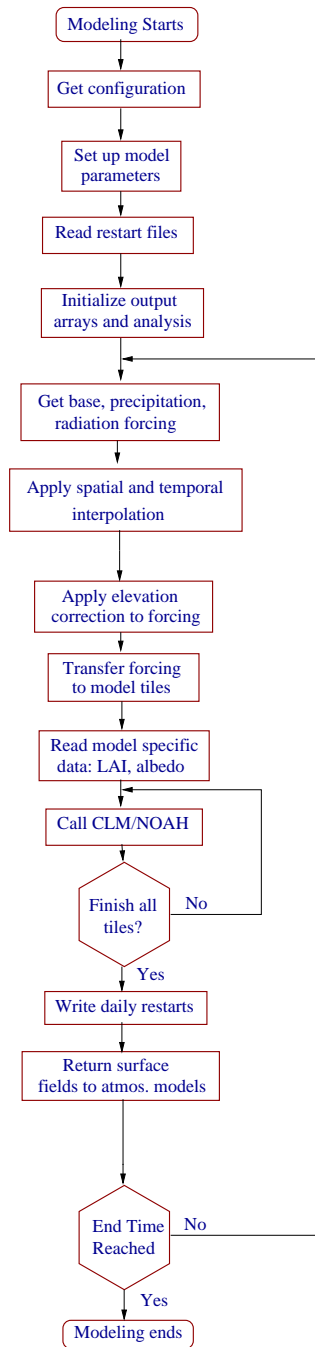


Figure 1: Flowchart for LDAS

2.3 Community Land Model (CLM)

CLM is a 1-D land surface model, written in Fortran 90, developed by a grass-roots collaboration of scientists who have an interest in making a general land model available for public use. LDAS currently uses CLM version 1.0, formerly known as common land model. CLM version 2.0 was released in May 2002 and will be implemented in LDAS in future. The source code for CLM 2.0 is freely available from the National Center for Atmospheric Research (NCAR) (<http://www.cgd.ucar.edu/tss/clm/>). The CLM is used as the land model for the community climate system model (CCSM) (<http://www.cesm.ucar.edu/>) and the community atmosphere model (CAM) (<http://www.cgd.ucar.edu/cms/>). CLM is executed with all forcing, parameters, dimensioning, output routines, and coupling performed by an external driver of the user's design (in this case done by LDAS). CLM requires pre-processed data such as the land surface type, soil and vegetation parameters, model initialization, and atmospheric boundary conditions as input. The model applies finite-difference spatial discretization methods and a fully implicit time-integration scheme to numerically integrate the governing equations. The model subroutines apply the governing equations of the physical processes of the soil-vegetation-snowpack medium, including the surface energy balance equation, Richards' [7] equation for soil hydraulics, the diffusion equation for soil heat transfer, the energy-mass balance equation for the snowpack, and the Collatz et al. [3] formulation for the conductance of canopy transpiration.

2.4 The Community NOAH Land Surface Model

The community NOAH Land Surface Model is a stand-alone, uncoupled, 1-D column model freely available at the National Centers for Environmental Prediction (NCEP; <ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noah1sm/>). NOAH can be executed in either coupled or uncoupled mode. It has been coupled with the operational NCEP mesoscale Eta model [1] and its companion Eta Data Assimilation System (EDAS) [8], and the NCEP global Medium-Range Forecast model (MRF) and its companion Global Data Assimilation System (GDAS). When NOAH is executed in uncoupled mode, near-surface atmospheric forcing data (e.g., precipitation, radiation, wind speed, temperature, humidity) is required as input. NOAH simulates soil moisture (both liquid and frozen), soil temperature, skin temperature, snowpack depth, snowpack water equivalent, canopy water content, and the energy flux and water flux terms of the surface energy balance and surface water balance. The model applies finite-difference spatial discretization methods and a Crank-Nicholson time-integration scheme to numerically integrate the governing equations of the physical processes of the soil vegetation-snowpack medium, including the surface energy bal-

ance equation, Richards' [7] equation for soil hydraulics, the diffusion equation for soil heat transfer, the energy-mass balance equation for the snowpack, and the Jarvis [4] equation for the conductance of canopy transpiration.

3 Description of the Test Case

The global land surface is modeled by dividing it into two-dimensional regions or cells (e.g. cells of size 1km x 1km, which would lead to approximately 50,000 times more grid points than that of LDAS with cells of size $2^\circ \times 2.5^\circ$). Each cell can have a partial spatial coverage by a number of vegetation types, as well as bare soil. The vegetation characteristics such as leaf area index, stomatal resistance, etc. might be time varying. The conditions in each cell (energy, water fluxes, etc.) are computed at different time intervals. Each cell is driven by different atmospheric forcing variables.

Assuming approximately 0.4 milliseconds for each LSM run on a particular cell, it can be estimated that modeling land surface processes over a year with 15 minute timesteps would require approximately 74 years of runtime. This problem is clearly a grand challenge simply from computational perspective.

The baselining results presented in this report were obtained by executing the LDAS system on the following NASA AMES systems.

- HOPPER: SGI Origin 2000 IRIX64 6.5, 64 250MHz IP27 Processors
- LOMAX: SGI Origin 3000 IRIX64 6.5, 512 400MHz IP35 Processors

The domain resolution was set to be $1/4^\circ$. The CLM and NOAH LSMs were used in various runs. Two different timesteps (15 and 30 minutes) were used in all runs. For simplicity, only one tile per grid was simulated in the runs. The output files were written using the GRIB format. Various combinations of forcings (including precipitation, shortwave, longwave radiation) were employed in the runs. The scalability of the code on a single processor at different domain resolutions was also examined.

4 Description of the Computer Code Used

This section provides an algorithmic description of the computer code used in the baselining. LDAS is a model control and input/output system (consisting of a number of subroutines, modules written in Fortran 90 source code) that drives multiple offline one-dimensional LSMs using a vegetation defined "tile" or "patch" approach to simulate subgrid scale variability. The one-dimensional LSMs, which are subroutines of LDAS, apply the governing equations of the physical processes of the soil-vegetation-snowpack medium. These equations are model independent.

4.1 Documentation of the Computer Code

The documentation of LDAS and the land surface models (CLM 1.0 and NOAH 2.5) can be accessed at <http://lis.gsfc.nasa.gov/docs/LDAS-Doc/ldas2/index.html>.

4.2 Code Repository

The computer source code employed in the baselining may be obtained from the LIS baseline repository .

5 Results

LDAS was run on different SGI Origin systems with various combinations of forcings and different land surface models. The simulated period of time considered for all the runs in this study is 1 day. The computational demands of various runs are quantified using four parameters: Total execution times, CPU times, disk usage, and memory usage.

5.1 Total Execution Times

The total execution time is a typical parameter used to evaluate the performance of a code. Figure 2 presents the total execution times for the LDAS runs using CLM and NOAH on HOPPER and LOMAX. The timestep for the simulations is 1800 seconds. In Figure 2, H represents HOPPER and L represents LOMAX. It can be observed that the execution times on LOMAX are considerably lower than those on HOPPER.

5.2 CPU Times

A dynamic runtime profiling, using SGI's speedshop toolkit, was conducted to identify the most computationally intensive features of the code. Figures 3, 4, 7, and 8 show the CPU times of these functions. They are identified as:

- **getgeos**: This function opens, reads, and interpolates GEOS (Goddard Earth Observing System) forcing.
- **ipolates**: This function performs spatial interpolation.
- **zterp**: This function computes the zenith angle based temporal interpolation.

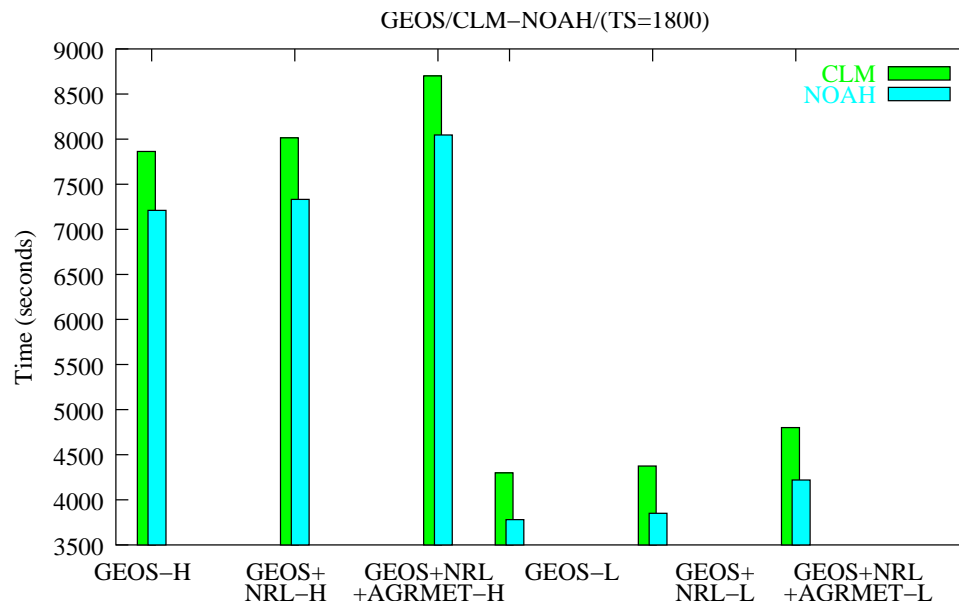


Figure 2: Total execution times for the LDAS runs on HOPPER and LOMAX.

- **getgrad**: This function opens, reads, interpolates, and overlays radiation forcing.
- **getglbpcp**: This function opens and reads global precipitation forcing.
- **clm_main**: This is the main call to the CLM LSM.
- **clm_out**: CLM output writer.
- **noah_main**: This is the main call to the NOAH LSM.
- **noah_out**: NOAH output writer.

The corresponding contributions to the CPU times are shown as percentages in Figures 5, 6, 9 and 10. GEOS, NRL, and AGRMET indicate GEOS forcing, NRL precipitation forcing, and AGRMET radiation forcings, respectively. TS=1800 denote a timestep of 30 minutes.

It can be observed that the contributions of functions in terms of percentages remain consistent across different computers, although the actual computational times differ.

The impact of the duration of timestep was examined by reducing the timestep from 30 minutes to 15. The results obtained using CLM on HOPPER is shown as a representative sample in Figure 11. The values obtained at 15 minute timesteps

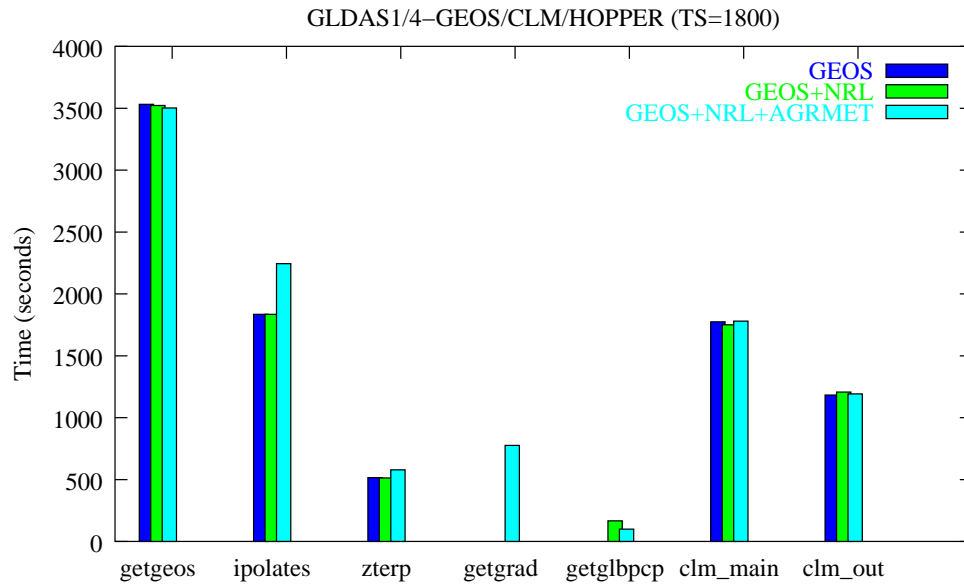


Figure 3: CPU times of computationally intensive functions on HOPPER for CLM runs

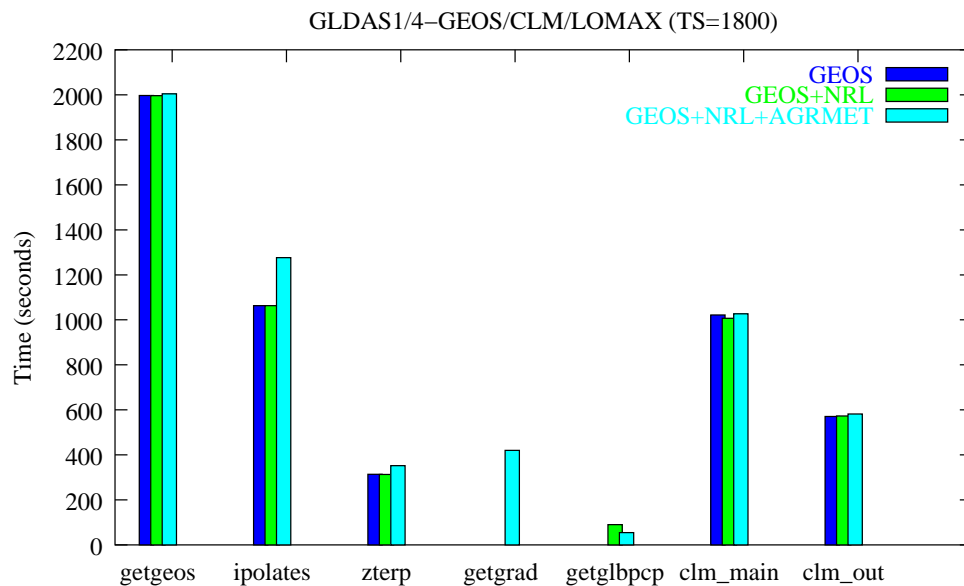


Figure 4: CPU times of computationally intensive functions on LOMAX for CLM runs

are compared with projected values using the computational times obtained using 30 minute timesteps. It can be observed that the time for the LSM calls increase (2

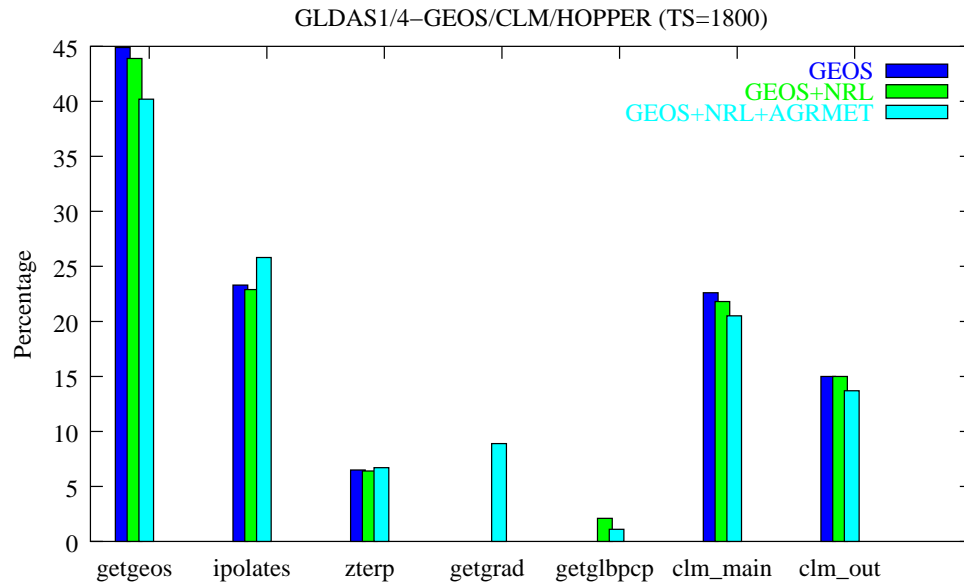


Figure 5: Percentage of total CPU times for computationally intensive functions on HOPPER for CLM runs

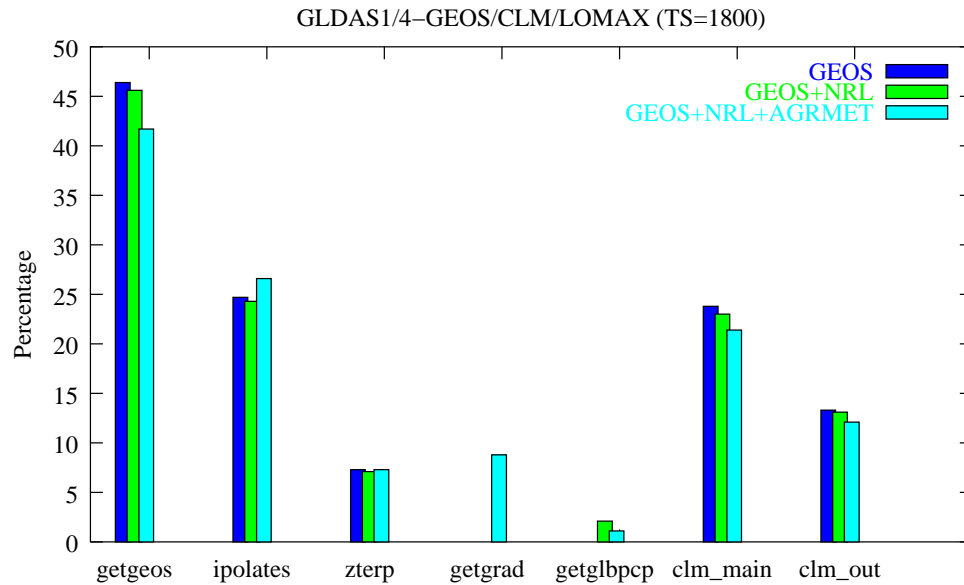


Figure 6: Percentage of total CPU times for computationally intensive functions on LOMAX for CLM runs

fold) with reduction in timestep duration. The interpolation functions (**ipolates** and **zterp**) are not affected by the decreased timestep since the data is not interpolated at

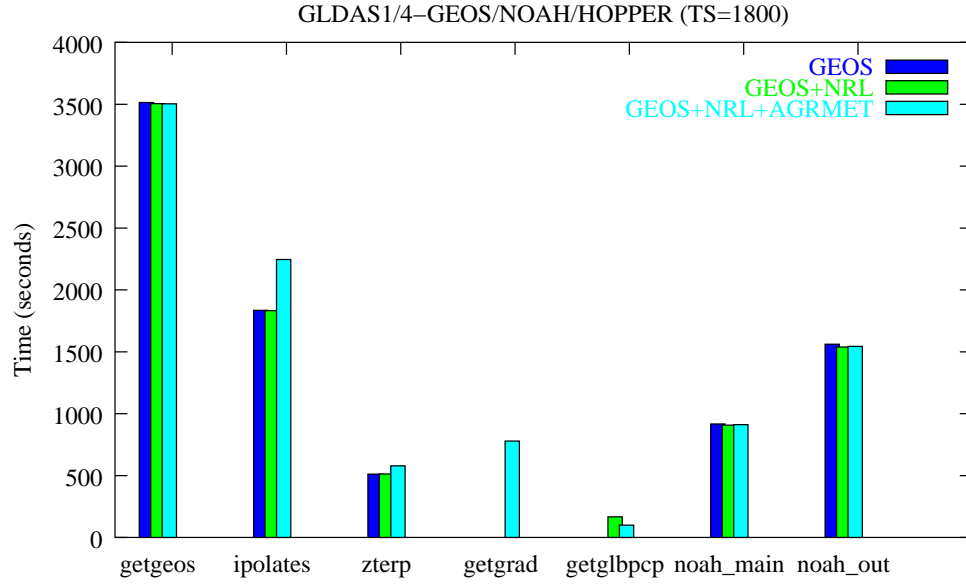


Figure 7: CPU times of computationally intensive functions on HOPPER for NOAH runs

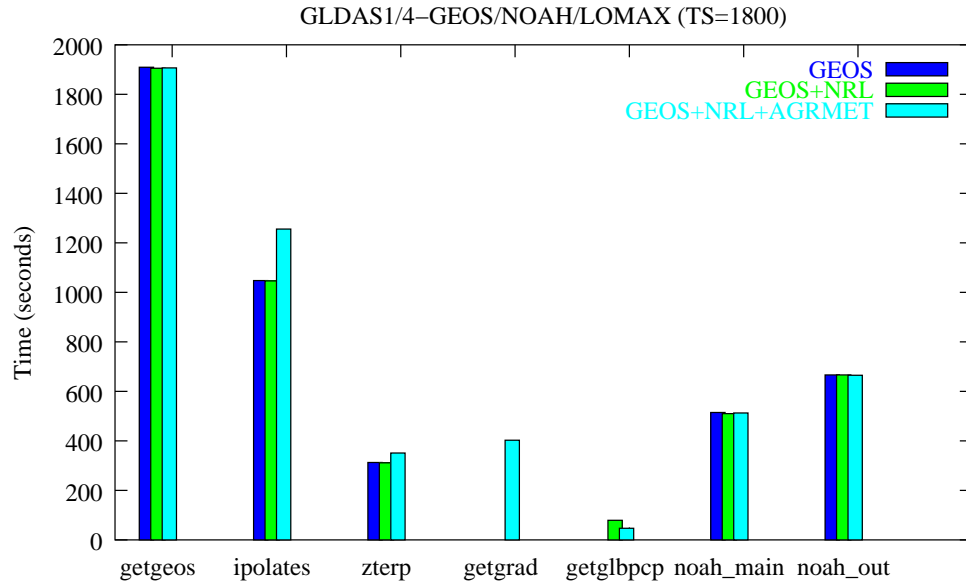


Figure 8: CPU times of computationally intensive functions on LOMAX for NOAH runs

every timestep. This leads to the less than linear scaling of the main forcing function (getgeos).

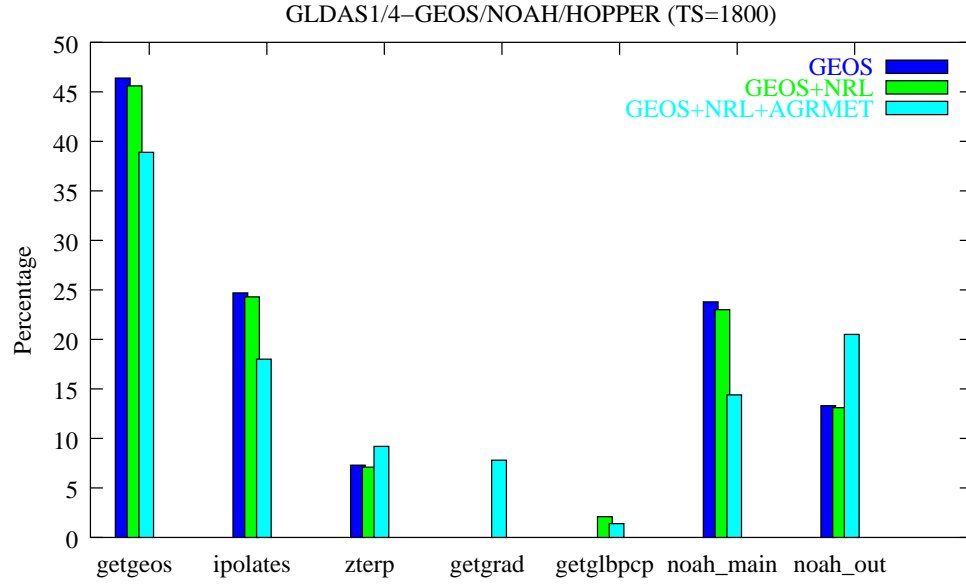


Figure 9: Percentage of total CPU times for computationally intensive functions on HOPPER for NOAH runs

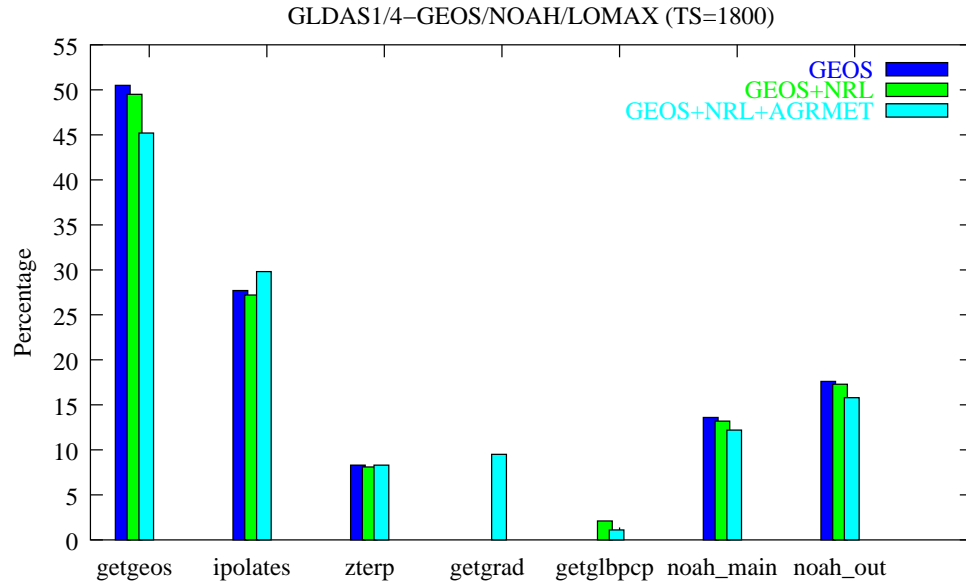


Figure 10: Percentage of total CPU times for computationally intensive functions on LOMAX for NOAH runs

To study the scalability of the code with increase in domain size, profiling studies were conducted for a domain increase from $2^\circ \times 2.5^\circ$ to $1/4^\circ$. The results are shown

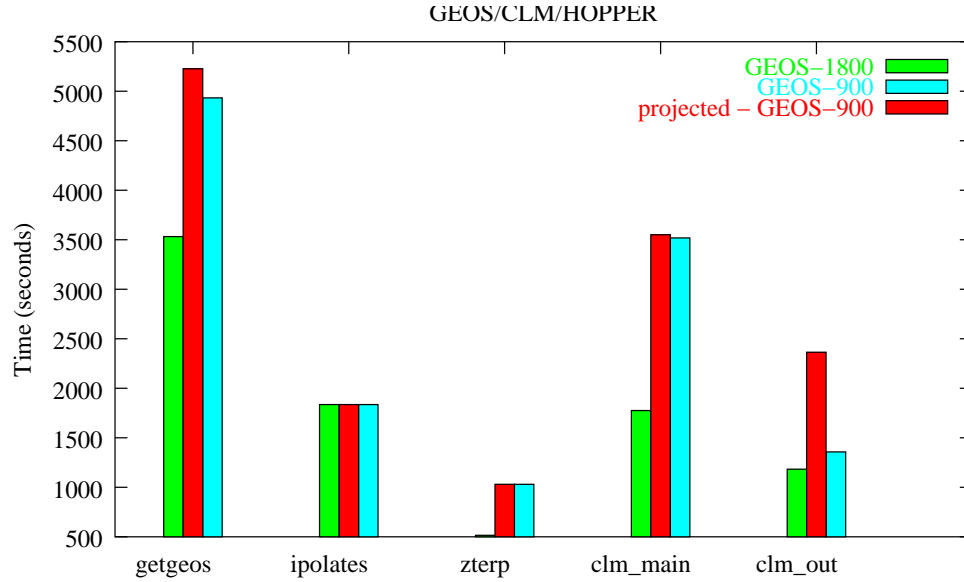


Figure 11: Effect of timestep duration on the computational times for CLM runs

in Figures 12 and 13 for runs on HOPPER. The computational times at $1/4^\circ$ are also compared with some projected values computed from the $2^\circ \times 2.5^\circ$ runs. It can be observed that the different segments of the code scales mostly as expected with the output routines of the LSMs being the notable exceptions. This could be attributed to the nonlinear scaling nature of the specific output libraries in these routines.

Table 1 lists a measure of computational intensity on different platforms for various runs. It can be observed that the performance of the code on LOMAX is significantly better than that on HOPPER. For example, the computational complexity measured for the CLM run using GEOS forcing on HOPPER is approximately twice that of the run on LOMAX. It can also be seen that CLM is computationally more intensive than NOAH. The measured computational complexities for NOAH are smaller than that of CLM as shown in Table 1. From the Figures 3, 4, 7, and 8, it can also be observed that the calls to CLM routines take more time than those of NOAH.

5.3 Disk Usage

The LDAS code uses three categories of global data: parameter data, input forcing data and output data. The parameter data include vegetation classification, land mask, etc., with a size of approximately 136GB. The code reads in the forcing data at regular intervals, with the traffic estimated to be approximately 279 MB/day.

For the baselining results presented in this report, the code and requisite files require 1.1GB of hard disk space. The disk space required for output for different

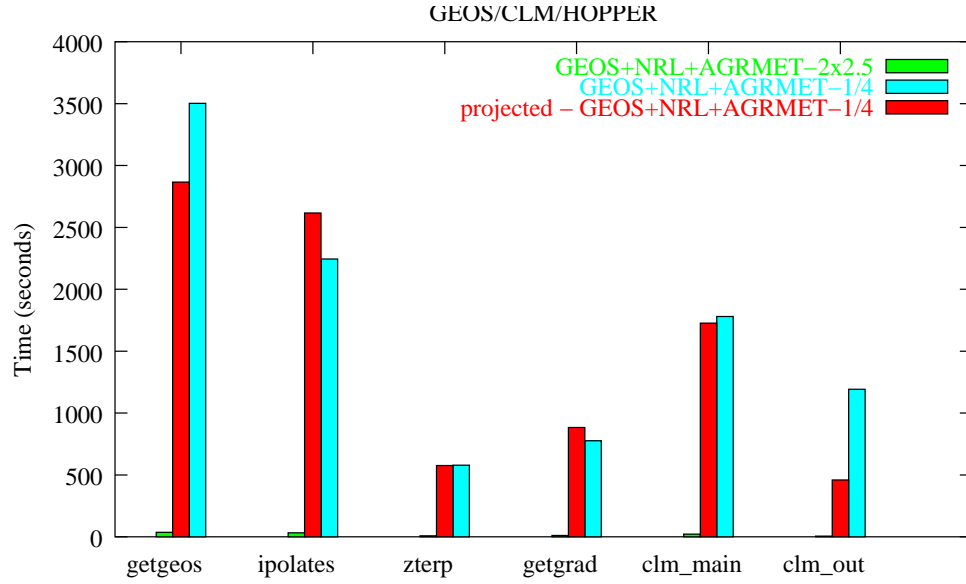


Figure 12: Effect of domain increase from $2^\circ \times 2.5^\circ$ to $1/4^\circ$ on the computational times for CLM runs

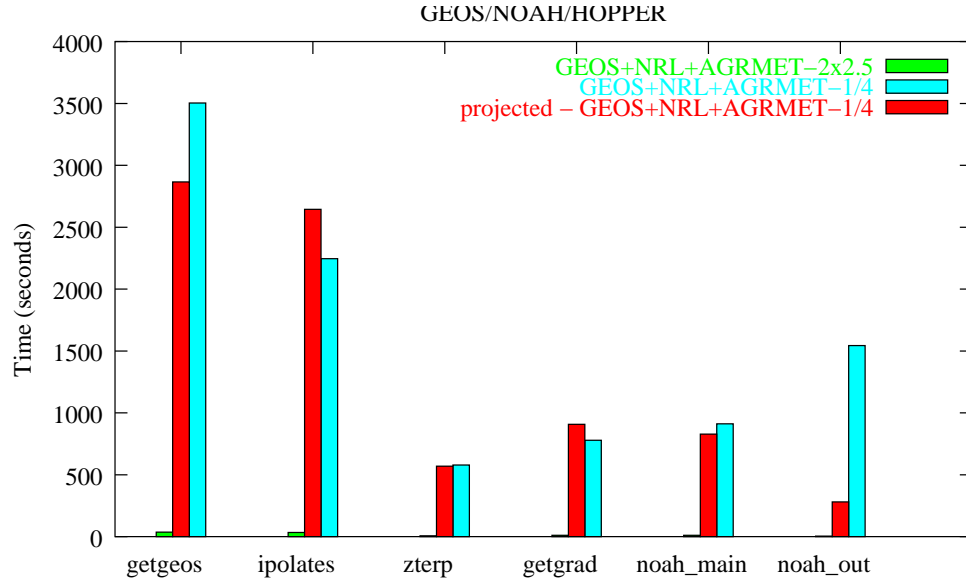


Figure 13: Effect of domain increase from $2^\circ \times 2.5^\circ$ to $1/4^\circ$ on the computational times for NOAH runs

baselining runs are shown in Table 2. It can be noticed that the disk usage increases almost linearly with the increase in domain size.

Table 1: Measure of computational intensity for LDAS 1/4° runs (ms/gridcell/day)

		CLM		NOAH	
		Timestep (minutes)			
		15	30	15	30
HOPPER	GEOS	13.9	9.1	13.7	8.3
	GEOS + NRL	14.2	9.3	14.1	8.5
	GEOS + NRL + AGRMET	15.2	10.1	14.6	9.3
LOMAX	GEOS	7.7	5.0	6.9	4.4
	GEOS + NRL	7.9	5.1	7.1	4.7
	GEOS + NRL + AGRMET	8.4	5.6	7.6	4.9

Table 2: Disk Usage for various LDAS runs (in MB)

	CLM	NOAH
LDAS 2° × 2.5°	5	3
LDAS 1/4°	400	235

5.4 Memory Usage

The LDAS code also requires significant memory for execution. The following table 3 lists the approximate memory requirements for LDAS runs with different land surface models.

Table 3: Memory Usage for various LDAS runs

	CLM	NOAH
LDAS 2° × 2.5°	250 MB	200 MB
LDAS 1/4°	3.5 GB	2.0 GB

5.5 Computational Requirements at 1km

The baselining results presented above are useful in making calculated projections on the computational requirements when LDAS is run at 1km resolution. As mentioned

earlier, in going from domain of a lower resolution to one of higher, the number of grid cells increases. For example, at 1km resolution, there will be 80 times more grid cells at $1/4^\circ$ and 50,000 times more grid cells than at $2^\circ \times 2.5^\circ$. The land surface modeling in LDAS involves calls to the land surface model for each grid cell. As a result, the time required for the land surface model runs are expected to increase linearly with increase in grid points. Further, spatial interpolation routines such as `getgeos`, `getgrad`, etc., are also expected to scale linearly with increase in domain points. Although some segments of the code (such as the output routines) did not scale linearly with increase in grid cells, a rough estimate of the required total execution time can be estimated by linearly extrapolating with respect to the grid cells. To simulate a period of 1 day, the required computational time is approximately:

$$\begin{aligned} \text{Total Execution Time (at 1km with CLM)} &= 80,000 \times 92 \\ &\approx 85 \text{ days} \end{aligned}$$

$$\begin{aligned} \text{Total Execution Time (at 1km with NOAH)} &= 80,000 \times 83 \\ &\approx 77 \text{ days} \end{aligned}$$

These values are based on the $2^\circ \times 2.5^\circ$ runs on HOPPER.

In addition to the computing time, the disk and memory usage requirements also increase significantly at 1km. Using the values presented in Table 2, the output data volume for the global 1km run using CLM and NOAH can be estimated to be approximately 250 GB/day and 150 GB/day, respectively. Similarly, using these values from Table 3, the memory requirements for LDAS run at 1km resolution on a single machine can be estimated to be approximately 2TB.

6 Conclusions

These profiling results have demonstrated the functions that are most time-consuming, thereby identifying the portions of our code-set that require our immediate attention. These profiling results also demonstrate that these critical functions scale with respect to time and space as predicted, suggesting that across-the-board performance improvements can be made from re-writing these critical routines, instead of having to make specialized performance improvements for specific scenarios. As discussed in the earlier section, Figure 11 shows the scalability of various code segments with timestep and Figures 12 and 13 shows the scalability of the code for a larger domain.

The baselining study has also helped in quantifying the disk and memory usage requirements of the LDAS code. As mentioned earlier, the estimated disk output volume at 1km resolution (150-250GB/day) is extremely large and it is not feasible to store the output in a single file. As a result, the computing strategy must involve

a design to distribute the data across different nodes to keep the output data volume manageable. Similarly, the projected memory usage at 1km resolution is very large ($\sim 2\text{TB}$) and the problem need to be split up into smaller pieces to satisfy the memory requirements of a real-time operation.

7 Future Directions

In order to meet future milestones F and G, regarding the performance of LIS, we need to be able to characterize the behavior of our initial code set and use this characterization to guide our software development, system design, and code improvement. From the results presented in this report, it is apparent that global scale land surface modeling at 1km resolution poses significant computational challenges, from a computational as well as data/memory management perspectives. Parallel computing has emerged as the enabling technology that will help modern computers satisfy increasing high performance computing requirements. We plan to build a system that takes advantage of scalable parallel computing technologies to facilitate global land surface modeling. The land surface processes have rather weak horizontal coupling on short time and large space scales, enabling highly efficient scaling across massively parallel computing platforms. The high data densities could pose limitations on the land surface modeling efficiencies, and the LIS system will explore the use of high performance technologies to eliminate this bottleneck.

References

- [1] F. Chen, K. Mitchell, J. Schaake, Y. Xue, H. Pan, V. Koren, Y. Duan, M. Ek, and A. Betts. Modeling of land-surface evaporation by four schemes and comparison with fife observations. *J. Geophys. Res.*, 101(D3):7251–7268, 1996.
- [2] CLM. <http://www.cgd.ucar.edu/tss/clm/>.
- [3] G. J. Collatz, C. Grivet, J. T. Ball, and J. A. Berry. Physiological and environmental regulation of stomatal conductance: Photosynthesis and transpiration: A model that includes a laminar boundary layer. *Agric. For. Meteorol.*, 5:107–136, 1991.
- [4] P. G. Jarvis. The interpretation of leaf water potential and stomatal conductance found in canopies of the field. *Phil. Trans. R. Soc.*, B(273):593–610, 1976.
- [5] LDAS. <http://ldas.gsfc.nasa.gov>.

- [6] NOAA. <ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noahsm/>.
- [7] L. A. Richards. Capillary conduction of liquids in porous media. *Physics*, 1:318–333, 1931.
- [8] E. Rogers, T. L. Black, D. G. Deaven, G. J. DiMego, Q. Zhao, M. Baldwin, N. W. Junker, and Y. Lin. Changes to the operational "early" eta analysis/forecast system at the national centers of environmental prediction. *Wea. Forecasting*, 11:391–413, 1996.